Page Modification Logging for Virtual Machine Monitor White Paper

This document is intended only for VMM or hypervisor software developers and not for application developers or end-customers. Readers are expected to be knowledgeable about Intel® Architecture and Intel® Virtualization Technology.

331872-001

January 2015

Intel technologies features and benefits depend on system configuration and may require enabled hardware, software, or service activation. Learn more at intel.com, or from the OEM or retailer.

No computer system can be absolutely secure. Intel does not assume any liability for lost or stolen data or systems or any damages resulting from such losses.

You may not use or facilitate the use of this document in connection with any infringement or other legal analysis concerning Intel products described herein. You agree to grant Intel a non-exclusive, royalty-free license to any patent claim thereafter drafted which includes subject matter disclosed herein.

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

The products described may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

This document contains information on products, services and/or processes in development. All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest Intel product specifications and roadmaps

Copies of documents which have an order number and are referenced in this document, or other Intel literature, may be obtained by calling 1-800-548-4725, or by visiting http://www.intel.com/design/literature.htm.

Intel, the Intel logo, Intel Atom, Intel Core, Intel SpeedStep, MMX, Pentium, VTune, and Xeon are trademarks of Intel Corporation in the U.S. and/or other countries.

*Other names and brands may be claimed as the property of others.

CHAPTER 1
PAGE MODIFICATION LOGGING

CHAPTER 1
PAGE MODIFICATION LOGGING

1.1 OVERVIEW

This paper describes an Intel® Virtualization Technology (Intel® VT) enhancement for future Intel processors. This feature, referred to as **page-modification logging** (**PML**), further extends the capability of virtual-machine monitor (VMM) software that employs the extended page-table mechanism (EPT) by allowing that software to monitor the guest-physical pages modified during guest virtual machine (VM) execution more efficiently. Details of Intel VT, including EPT, can be found in *Intel® 64 and IA-32 Architectures Software Developer's Manual* (SDM) in Volume 3C.

Earlier Intel processors introduced accessed and dirty flags for EPT; see Section 28.2.4, "Accessed and Dirty Flags for EPT," of *Intel® 64 and IA-32 Architectures Software Developer's Manual* (SDM) in Volume 3C. This feature enables VMMs to implement memory-management and page-classification algorithms efficiently so as to optimize VM memory operations such as defragmentation, paging, etc. Without accessed and dirty flags, VMMs may need to emulate them (by marking EPT paging-structures as not-present or read-only) and incur the overhead of the resulting EPT violations: VM exits and associated software processing.

For some usage models, VMMs may benefit from additional support to monitor the working set size. As accessed and dirty flags for EPT are set without invoking the VMM, there is no opportunity for the VMM to accumulate working-set statistics during operation. To calculate such statistics the VMM must scan the EPT paging structures to aggregate the information from the accessed and dirty flags. Such scans impose both latency and bandwidth costs that may be unacceptable in some circumstances.

PML builds upon the processor's support for accessed and dirty flags for EPT, extending the processing that occurs when dirty flags for EPT are set. When PML is active each write that sets a dirty flag for EPT also generates an entry in an in-memory log, reporting the guest-physical address of the write (aligned to 4 KBytes). When the log is full, a VM exit occurs, notifying the VMM. A VMM can monitor the number of pages modified by each thread by specifying an available set of log entries.

The rest of this paper is organized in subsections which cover specific changes in VMX to support PML:

- Section 1.2 details changes to the VMCS introduced with the PML feature.
- Section 1.3 details changes to VMX non-root operation.
- Section 1.4 describes changes to VM entries.
- Section 1.5 describes changes to VM exits.
- Section 1.6 details changes to the capability reporting introduced with PML.

1.2 VMCS CHANGES

Secondary processor-based VM-execution control 17 is defined as **enable PML**.

A new 64-bit VM-execution control field is defined called the **PML address**. This is the 4-KByte aligned physical address of the **page-modification log**. The page-modification log comprises 512 64-bit entries. The VMCS-field encoding pair for the PML address is 0000200EH (for all 64 bits) and 0000200FH (for the upper 32 bits).

A new 16-bit guest-state field is defined called the **PML index**. The PML index is the logical index of the next entry in the page modification log. Because the page-modification log comprises 512 entries (see above), the PML index is typically a value in the range 0–511. The VMCS-field encoding for the PML index is 00000812H.

The PML address and PML index fields exist only on processors that support the 1-setting of the “enable PML” VM-execution control.

1.3 CHANGES TO VMX NON-ROOT OPERATION

If the “enable PML” VM-execution control is 1 and bit 6 of EPT pointer (EPTP) is 1 (enabling accessed and dirty flags for EPT), the behavior in VMX non-root operation is modified as described in this section[1].

Before allowing a guest-physical access, the processor may determine that it first needs to set an accessed or dirty flag for EPT. When this happens, the processor examines the PML index. If the PML index is not in the range 0–511 (bits 15:9 of PML index are not all 0), a VM exit occurs. The accessed or dirty flag is not set, and the guest-physical access that triggered the event does not occur. See Section 1.5 for details of how the VM exit occurs.

If instead the PML index is in the range 0–511 and a guest-physical access causes the processor to update a dirty flag for EPT (changing it from 0 to 1), the processor operates as follows:

- The guest-physical address of the access is written to the page-modification log. Specifically, the guest-physical address is written to physical address determined by adding 8 times the PML index to the PML address. Bits 11:0 of the value written are always 0 (the guest-physical address written is 4-KByte aligned).
- The PML index is decremented by 1 (this may cause the value to transition from 0 to FFFFH).

Because the processor decrements the PML index with each log entry, the value will eventually wrap around to FFFFH. No further logging will occur because, the next time a log entry is required, the processor will determine that bits 15:9 of the PML index are not all 0 and will cause a VM exit (see above and Section 1.5).

[1] If bit 6 of EPT pointer (EPTP) is 0 (disabling accessed and dirty flags for EPT), the setting of the “enable PML” VM-execution control has no effect on VMX non-root operation.

The following pseudocode comprehends the impact of the 1-setting of “enable PML” on an update to an accessed or dirty flag for EPT:

```
IF (PML Index[15:9] ≠ 0)
    THEN VM exit;
FI;
set accessed and dirty flags for EPT;
IF (a dirty flag was updated from 0 to 1)
    THEN
         PML address[PML index] ← 4-KByte-aligned guest-physical address;
         PML index is decremented;
FI;
```

1.4 CHANGES TO VM ENTRIES

If the “activate secondary controls” and “enable PML” VM-execution controls are both 1, VM entries ensure the following:

- The “enable EPT” VM-execution control is 1.
- Bits 11:0 of the PML address are 0.
- The PML address does not set any bits beyond the processor’s physical-address width.[1]

VM entry fails if this check fails. When such a failure occurs, control passes to the next instruction, RFLAGS.ZF is set to 1 to indicate the failure, and the VM-instruction error field is loaded with value 7, indicating “VM entry with invalid control field(s).”

This check may be performed in any order with respect to other checks on VMX controls and the host-state area. Different processors may thus give different error numbers for the same VMCS.

The “enable PML” VM-execution control may be 1 even if bit 6 of EPT pointer (EPTP) is 0 (disabling accessed and dirty flags for EPT); VM entry will not fail because of this condition. As noted in footnote 1 in Section 1.3, the setting of the “enable PML” VM-execution control has no effect on VMX non-root operation if accessed and dirty flags for EPT are disabled.

VM entry does not check the value of the PML index, and that value will not cause VM entry to fail, even if bits 15:9 of the value are not all 0.

[1] Software can determine a processor’s physical-address width by executing CPUID with 80000008H in EAX. The physical-address width is returned in bits 7:0 of EAX.

1.5 CHANGES TO VM EXITS

As indicated in Section 1.3, the processor causes a VM exit when the next PML index specifies an invalid location (because bits 15:9 of PML index are not all 0). Typically, this will occur when the page-modification log is full and the PML index has wrapped around from 0 to FFFFH.

VM exits resulting from the value of the PML index use a basic exit reason of 62 (3EH), indicating “page-modification log full.”

VM exits due to “page-modification log full” do save an exit qualification defined as follows:

- Bits 11:0 are undefined.
- Bit 12 is undefined in either of the following cases:
  - If the “NMI exiting” VM-execution control is 1 and the “virtual NMIs” VM-execution control is 0.
  - If the VM exit sets the valid bit in the IDT-vectoring information field.

  Otherwise, bit 12 is defined as follows:
  - If the “virtual NMIs” VM-execution control is 0, the EPT violation was caused by a memory access as part of execution of the IRET instruction, and blocking by NMI was in effect before execution of IRET, bit 12 is set to 1.
  - If the “virtual NMIs” VM-execution control is 1,the EPT violation was caused by a memory access as part of execution of the IRET instruction, and virtual-NMI blocking was in effect before execution of IRET, bit 12 is set to 1.
  - For all other relevant VM exits, bit 12 is cleared to 0.
- Bits 63:13 are undefined.

These VM exits save the IDT-vectoring information and IDT-vectoring error code as they are for VM exits due to EPT violations. Thus, if the VM exit occurred while delivering an event through the IDT, these fields receive information about that event.

1.6 CHANGES TO VMX CAPABILITY REPORTING

Section 1.3 specified that bit 17 of the secondary processor-based VM-execution controls is defined as “enable PML”. A processor that supports the 1-setting of “enable PML” sets bit 49 of the IA32_VMX_PROCBASED_CTLS2 MSR (index 48BH). RDMSR of that MSR returns 1 in bit 17 of EDX.